テンポラル・ロジックで用いられている連接について(アルゴリズムの数学的基礎理論とその応用)

木村, 晋二; 矢島, 脩三

数理解析研究所講究録
591 巻 1986 年 278-287
610

テンポラル·ロジックで用いられている連接について

木村 晋二 (神戸大学工学部), 矢島 脩三 (京都大学工学部)

On Concatenations Used in Temporal Logic

Shinji KIMURA (Kobe University) and Shuzo YAJIMA (Kyoto University)

あらまし:ここでは,命題論理の拡張である命題テンポラル·ロジック,命題テンポラル·ロジックの拡張である拡張テンポラル·ロジック,およびインターバル·テンポラル·ロジックについて,それらの論理式と系列集合の対応関係について考察し,テンポラル·ロジック固有の演算である○(next),□(henceforth)などが,系列集合上の一般化連接(系列の直後に系列を並べる通常の連接とは異なる)に対応していることを示す.また,テンポラル·ロジック上の判定問題を系列集合上の判定問題に帰着し,その決定可能性についても述べる.

1. まえがき

近年,時間を扱える論理体系として,命題論理の拡張であるテンポラル·ロジック(Temporal Logic, 以下ではTLと略す)が注目され,並行プログラムや論理回路などの動作記述および検証に用いられている(1)-(7).テンポラル·ロジックにおける論理式の真偽は,あるアルファベット上の系列に対して定義されるが,論理式とそれを真にする系列集合の対応関係が直接扱われることは少なく,とくにテンポラル·ロジックで用いられている論理演算が系列集合上のどのような演算に対応しているかはほとんど明らかにされていなかった.そこでここでは,命題テンポラル·ロジック(Propositional TL, 以下ではPTLと略す)(3),命題テンポラル·ロジックの拡張である拡張テンポラル·ロジック(Extended TL, 以下ではETLと略す)(4),インターバル·テンポラル·ロジック(Interval TL, 以下ではITLと略す)(6)におけるこれらの対応関係について考察する.

テンポラル·ロジックにおける論理式は,原子命題および論理演算から構成される.よって,論理式に対応する系列集合は,論理式の真偽値の定義に違反しないように,原子命題に対応する系列集合および論理演算に対応する系列集合上の演算を決めれば一意に決まる.

ここではまず原子命題と系列集合の対応関係として,PTL, ETL, およびITLに制限を加えたLocal ITLでは,原子命題に対し長さ1の系列集合(正則集合)を対応させていることを示す.また,制限を加えないITLでは,任意の系列集合(正則集合以外の系列集合を含む)を対応させていることを示す.

つぎに,論理演算のうち,¬(not),∧(and)などの論理演算が,系列集合の補集合演算,共通集合演算の一般化に対応することを示す.また,○(next),□(henceforth)などのテンポラル·ロジック固有の演算が,系列集合上の一般化連接(系列の直後に系列を並べる通常の連接とは異なる)に対応することを示す.

テンポラル·ロジックにおける恒真性判定は,論理式を真にする系列集合を用いることにより,系列集合の包含判定に帰着される.ここではまず,テンポラル·ロジックで用いられている論理演算に対応する系列集合上の演算が,いずれも正則集合のクラスで閉じることを示す.このことから,PTL, ETL, Local ITLにおける真偽判定が決定可能であることが自然に導ける.また,制限を加えないITLでは,原子命題に正則集合以外の系列集合を対応づけることができるために,真偽判定が決定不能となることが導ける.

本稿の2章では,基本的な定義を述べる.3章では,PTL, ETL, ITLおよびLocal ITLの論理式とその解釈について述べる.4章では,テンポラル·ロジックで用いられている論理演算に対応する系列集合上の演算について述べる.5章では,テンポラル·ロジック上の判定問題と系列集合上の判定問題の対応関係および判定問題の決定可能性について述べる.6章はむすびである.

2. 準備

アルファベット,系列,正則集合に関する基本的な定義は文献(8)に従う.

アルファベットΣ上の系列$x=a_1a_2\cdots a_n$ $(a_i\in\Sigma)$に対し,nをxの長さと呼び,$|x|$と表わす.また,xの部分系列$a_ia_{i+1}\cdots a_j$ $(1\leqq i\leqq j\leqq n)$を$x[i,j]$と表わす.$i>j$のときは,$x[i,j]=\varepsilon$とする.一記号からなる部分系列$a_i(1\leqq i\leqq n)$は,$x[i]$と表わす.

系列集合に対する商演算を定義する[8].アルファベットΣ上の系列xとyに対し,$x=yz$となるような$z(\in\Sigma^*)$が存在するとき,yに関するxの左からの商$y\backslash x$をzとする.また,そのようなzが存在しないときには$y\backslash x=\emptyset$とする.なお,任意の系列yに対し,$y\backslash\emptyset=\emptyset$とする.系列集合X, Yに対し,Yに関するXの左からの商$Y\backslash X$を,$\cup_{x\in X, y\in Y}\{y\backslash x\}$により定義する.また,系列yに関する系列xの右からの商x/yを,左からの商と同様に定義する.

右線形文法とは,非終端記号の集合N,終端記号の集合Σ,生成規則の集合$P(\subseteq N\times(\Sigma^*\cup\Sigma^*N))$,開始記号$S(\in N)$からなる4項組$(N,\Sigma,P,S)$である.以下では,生成規則$(V,y)$ $(V\in N, y\in(\Sigma^*\cup\Sigma^*N))$を$V\rightarrow y$のように表わす.

3. テンポラル·ロジックの論理式とその解釈

ここではまず,命題テンポラル·ロジック(PTL)[3],PTLを拡張した拡張テンポラル·ロジック(ETL)[4],インターバル·テンポラル·ロジック(ITL)[6]およびITLに制限を加えたLocal ITL[6]の論理式とその解釈について述べる.ここでの定義は基

本的には各参考文献に従うが,PTLやETLにおける論理式の真偽が通常無限系列上で定義されるのに対し,ここでは有限系列上で論理式の真偽を定義する.このとき問題となるのは,例えば○P(つぎの時間でPが成立する)の真偽値を長さ1の系列に対してどのように定義するかである.ここでは,真偽値を真(T),偽(F),未定義(⊥)の3値とすることにより対処している.

3.1 命題テンポラル·ロジック

命題テンポラル·ロジック(PTL)の論理式とその解釈を定義する.原子命題記号の有限集合をΔとすると,Δ上のPTLにおける論理式はつぎのように定義される.

(1) Δの要素pは論理式である.

(2) P, Qが論理式であるとき,¬PおよびP∧Qは論理式である.

(3) P, Qが論理式であるとき,○P, □P, P U Qは論理式である.

(4) 上記(1),(2),(3)の有限回の適用により構成されるもののみが論理式である.

これらの論理式の直感的な意味について述べる.命題論理の場合と同様,¬PはPが成立しないことを,またP∧QはP, Qの両方が成立することを表わす.○Pはつぎの時間でPが成立することを表わし,□Pは現在から将来ずっとPが成立することを表わす.またP U Qは,将来Qが成立するまでずっとPが成立することを表わす.

PTLの論理式Pの真偽値をここでは{T, F, ⊥(undefined)}の3値とする.Tは真を,Fは偽を,⊥は真でも偽でもないことを表わす.

Δ上のPTLのモデルとは,アルファベットΣと,Σ×Δから{T, F}への関数Mの二項組(Σ, M)である.Mは原子命題に真偽値を与える関数である.

PTLの論理式Pの真偽値は,Σ上の系列xに対し,つぎのように定義される.以下では,系列xに対する論理式Pの真偽値をx(P)と表わす.

(1) Δの要素pに対しては,

x(p)= M(x[1], p), ただし|x|≧1,
⊥, 上記以外,

とする.

(2) ¬Pに対しては,

x(¬P)= T, ただしx(P)=F,
F, ただしx(P)=T,
⊥, 上記以外,

とする.

(3) P∧Qに対しては,

x(P∧Q)= T, ただしx(P)=x(Q)=T,
⊥, ただしx(P)=⊥またはx(Q)=⊥,

F, 上記以外,

とする.

(4) $\bigcirc P$に対しては,

$x(\bigcirc P)=$ T, ただし$x[2, |x|](P)=T$,

F, ただし$x[2, |x|](P)=F$,

$\perp$, 上記以外,

とする.

(5) $\Box P$に対しては,

$x(\Box P)=$ T, ただしある$i(1 \leqq i \leqq |x|)$が存在し, i以下の任意のjに対して$x[j, |x|](P)=T$, かつiより大きい任意のkに対し$x[k, |x|](P)=\perp$,

$\perp$, ただし$x(P)=\perp$,

F, 上記以外,

とする.

(6) $P \cup Q$に対しては,

$x(P \cup Q)=T$, ただしある$i(1 \leqq i \leqq |x|)$が存在し, $x[i, |x|](Q)=T$かつiより小さい任意のjに対し$x[j, |x|](P)=T$, またはある$i(1 \leqq i \leqq |x|)$が存在し, iより小さい任意のjに対し$x[j, |x|](P)=T$かつi以上の任意のkに対し$x[k, |x|](P)=x[k, |x|](Q)=\perp$,

$\perp$, ただし$x(P)=x(Q)=\perp$,

F, 上記以外,

とする.

論理式の真偽値$\perp$に関しては, つぎの補題が成立する.

[補題1]

論理式Pに対し, ある非負整数i_Pがあり,

(1) i_Pよりも短い任意の系列xに対し, $x(P)=\perp$,

(2) i_P以上の任意の系列xに対し, $x(P)\neq\perp$,

である. □

3.2 拡張テンポラル・ロジック

PTLの拡張である拡張テンポラル・ロジック(ETL)の拡張演算の定義とその解釈を述べる. ETLでは, PTLで用いられている論理演算に加え, 右線型文法(V_N, V_T, P, S)の非終端記号$V_i\ (\in V_N)$に対応して一つの演算子G_iを定義している. いま$V_T=\{v_1, v_2, ..., v_n\}$であるとすると, 論理式$G_i(P_1, P_2, ..., P_n)$の直感的な意味は, 終端記号$v_i$に$P_i$を対応づけ, $V_i\ (\in V_N)$から生成されるV_T上の系列wに対し, 各時間kで$w[k]$に対応する論理式$P_{w[k]}$が成立するときにのみ$G_i(P_1, P_2, ..., P_n)$が成立するというものである. これを形式的に述べると, PTLと同様のモデル(Σ, M)のもとで,

Σ上の系列xに対し, $x(G_i(P_1, P_2, ..., P_n))=T$となるための必要十分条件は, 非終端記号$V_i$から$w=v_{j_1}v_{j_2}\cdots(1\leqq j_k\leqq n, v_{j_k}\in V_T)$が生成され, ある$i\,(1\leqq i\leqq|x|)$があり, i以下の任意のkに対し$x[k, |x|](P_{j_k})=T$であり, かつiより大きい任意のmに対し$x[m, |x|](P_{j_m})=\perp$であることである.

3.3 インターバル・テンポラル・ロジック

インターバル・テンポラル・ロジック(ITL)の論理式およびその解釈について述べる. 原子命題記号の有限集合をΔとすると, Δ上のITLにおける論理式はつぎのように定義される.

(1) Δの要素pは論理式である.
(2) P, Qが論理式であるとき, $\neg P$および$P\wedge Q$は論理式である.
(3) P, Qが論理式であるとき, $\bigcirc P$および$P\,;Q$は論理式である.
(4) 上記(1), (2), (3)の有限回の適用により構成されるもののみが論理式である.

$\neg P$, $P\wedge Q$, $\bigcirc P$の直感的な意味はPTLの場合と同様である. また$P\,;Q$は, Pが成立する時間間隔の後にQが成立する時間間隔が続くことを表わしている.

Δ上のITLのモデルとは, アルファベットΣと, $\Sigma^*\times\Delta$から$\{T, F\}$への関数Mの二項組(Σ, M)である. Mは原子命題に真偽値を与える関数である. Σ上の系列xに対し, ITLの論理式Pの真偽値$x(P)$はつぎのように定義される.

(1) Δの要素pに対しては,

$$x(p)=M(x, p)$$

とする.

(2) $\neg P$に対しては,

$$x(\neg P)=\begin{cases}T, & \text{ただし}\ x(P)=F,\\ F, & \text{ただし}\ x(P)=T,\end{cases}$$

とする.

(3) $P\wedge Q$に対しては,

$$x(P\wedge Q)=\begin{cases}T, & \text{ただし}\ x(P)=x(Q)=T,\\ F, & \text{上記以外},\end{cases}$$

とする.

(4) $\bigcirc P$に対しては,

$$x(\bigcirc P)=\begin{cases}T, & \text{ただし}\ x[2, |x|](P)=T,\\ F, & \text{ただし}\ x[2, |x|](P)=F,\end{cases}$$

とする.

(5) $P\,;Q$に対しては,

$x(P\,;Q)=T$, ただし, ある$i(1\leqq i\leqq|x|)$が存在し,

$x[1,i](P)=T$かつ$x[i,|x|](Q)=T$,

F, 上記以外,

とする.

ITLのモデルは,PTLのモデルにおけるMの定義域$\Sigma\times\Delta$を,$\Sigma^*\times\Delta$に拡張したものであるとみることができる.このため,ITLの論理式の真偽値は$\perp$とならない.

一方,Local ITLはMの定義域を$\Sigma\times\Delta$に制限したものである.よって,Local ITLのモデルはPTLのモデルに等しい.また,Local ITLの論理式の真偽値の定義は,PTLやETLと同様である.

4 テンポラル·ロジックの論理式に対応する系列集合

ここではPTL, ETL, Local ITLのモデル(Σ, M)に対し,論理式Pに対応するΣ上の系列集合S(P)を,Σ上の系列xに対し$x(P)=T$となるための必要十分条件が$x\underline{\in}S(P)$であるように定義する.ただし,$\underline{\in}$は通常の帰属関係の一般化であり,系列xおよび系列集合Xに対し,

(i) $x\in X$または

(ii) $\forall y\in X$に対し$|y|<|x|$, かつ$\exists y\in X$がありx=yz $(z\in\Sigma^*)$

であるとき,かつそのときにかぎり$x\underline{\in}X$とする.

また,ITLのモデル(Σ, M)に対し,論理式Pに対応するΣ上の系列集合S(P)を,Σ上の系列xに対し$x(P)=T$となるための必要十分条件が$x\in S(P)$であるように定義する.

4.1 原子命題に対応する系列集合

PTL, ETLおよびLocal ITLにおける原子命題pに対応する系列集合S(p)は,$S(p)=\{a\in\Sigma \mid M(a,p)=T\}$により定義する.また,ITLにおける原子命題pに対応する系列集合S(p)は,$S(p)=\{x\in\Sigma^* \mid M(x,p)=T\}$により定義する.この定義は,先に述べた条件を満たす.PTL, ETL, Local ITLではどのようなモデルを与えても,原子命題に対応する系列集合は正則集合である.ただし,ITLでは,モデルにより,原子命題に対応する系列集合が正則集合とならないことがある.

4.2 命題論理演算に対応する系列集合上の演算

ここでは,$\neg$および$\wedge$に対応する系列集合上の演算について考察する.まず$\neg$について述べる.PTL, ETLおよびLocal ITLの論理式Pに対し,$S(\neg P)$を$\{x\in\Sigma^* \mid$ S(P)中にある系列yが存在し$|x|\leqq|y|$かつ$x(P)=F\}$により定義する.例えば,原子命題pに対しては$S(\neg p)=\Sigma-S(p)$である.また,S(P)中に長さ最大の系列が存在しない

(すなわちいくらでも長い系列を含む)ときは, $S(\neg P)=\Sigma^* - S(P) - \{x \in \Sigma^* \mid x(P)=\perp\}$ である.

またITLでは, $S(\neg P)$ を $\Sigma^* - S(P)$ とする.

論理式PとQに対し, $S(P \wedge Q)$ を $\{x \in S(P) \cup S(Q) \mid x \underline{\in} S(P)$ かつ $x \underline{\in} S(Q)\}$ により定義する. これは通常の共通集合をとる演算($\cap$)の一般化であり, 原子命題p, qに対しては, $S(p \wedge q)=S(p) \cap S(q)$ が成立する.

またITLでは, $S(P \wedge Q)$ を $S(p) \cap S(q)$ とする.

4.3 テンポラル・ロジック固有の演算に対応する系列集合上の演算

ここでは, ○(next), □(henceforth), U(until), ETLにおける拡張演算GおよびITLにおける;(semicolon)の各論理演算に対応する系列集合上の一般化連接について述べる.

○に対しては, $\Sigma \cdot$ という一般化連接が対応する. すなわち, 論理式Pに対し, $S(\bigcirc P)$ を $\Sigma \cdot S(P)$ とする.

つぎに論理式□PおよびP U Qに対応する系列集合を与える前に, 一般化連接 $\textcircled{1}$ および $\textcircled{1}$ による閉包を定義する. 系列 $x, y\ (\in \Sigma^*)$ に対し, $x \,\textcircled{1}\, y$ を,

$$x \,\textcircled{1}\, y = \begin{cases} x[1] \cdot y, & \text{ただし} |x|=1, \text{または} |x| \geqq 1 \text{かつ} x[2, |x|]=y[1, |x|-1], \\ \emptyset, & \text{上記以外}, \end{cases}$$

とする. これは, 1記号ずらして等しい部分を重ねる演算である. なお, 一般化連接 $\textcircled{1}$ は結合則を満たさない.

系列集合Xの一般化連接を用いた閉包 $X^{\textcircled{1}}$ は, Kleene閉包と同様, 以下の(1)~(3)により定義される.

(1) $\varepsilon \in X^{\textcircled{1}}$. (2) $X \subseteq X^{\textcircled{1}}$. (3) $x, y \in X^{\textcircled{1}}$ ならば, $x \,\textcircled{1}\, y \in X^{\textcircled{1}}$.

さて, 論理式Pに対し, $S(\Box P)$ を $S'(P)^{\textcircled{1}}$ とする. ただし, $S'(P)$ は $\{x \in S(P) \mid$ 任意の $i\ (1 \leqq i \leqq |x|)$ に対し, $x[i, |x|](P)=T$ または $\perp\}$ を表わす. また, 論理式P, Qおよび $\$\ (\notin \Sigma)$ に対し, $S(P\ U\ Q)$ を $S'(P)^{\textcircled{1}} \cup ((S'(P) \cup S(Q) \cdot \$)^{\textcircled{1}} / \$ \cdot \Sigma^*)$ とする. ただし, $\cdot$ は通常の連接を, また $/$ は右からの商を表わす.

ETLにおける拡張演算Gにも, ある一般化連接による閉包が対応している. このときの一般化連接は, Gによって決まるある数kだけずらして重ね合わせる一般化連接 $\textcircled{1}_k$ と, 一般化連接 $\textcircled{1}$ を組み合わせたものである. すなわち, $V_i \to v_{i_1} v_{i_2} \cdots v_{i_k} V_{ij}\ (v_{i_m} \in V_T, V_{ij} \in V_N)$ に対しては, $(S(P_{i_1}) \,\textcircled{1}\, (S(P_{i_2}) \,\textcircled{1}\, (\cdots (S(P_{i_{k-1}}) \,\textcircled{1}\, S(P_{i_k})) \ldots) \,\textcircled{1}_k\, S(V_{ij})$ を対応づければ良い. ただし, $S(V_{ij})$ は V_{ij} に対応する演算子から生成される論理式に対応する系列集合を表わす. このように, ETLにおける拡張演算は, PTLにおける□に対し, ずらす記号の数の拡張になっている.

ITLにおける;には, 一般化連接∘が対応している. 一般化連接∘は, 系列xとyに対し,

$$x \circ y = \begin{cases} x[1, |x|-1]\cdot y, & \text{ただし } x[|x|] = y[1], \\ \varnothing, & \text{上記以外}, \end{cases}$$

により定義される.この一般化連接を,$_{\odot}$と同様に系列集合に対して拡張し,論理式P, Qに対して,S(P ; Q)をS(P)$\circ$S(Q)とする.

以上のように論理演算に対応する系列集合上の演算を定義したとき,つぎの定理が成立する.

[定理 1]

(1) PTL, ETL, Local ITLの論理式Pとある系列xに対し,x(P)=Tであるための必要十分条件は$x \in S(P)$である.

(2) ITLの論理式Pとある系列xに対し,x(P)=Tであるための必要十分条件は$x \in S(P)$である. □

5. テンポラル・ロジック上の判定問題の決定可能性

テンポラル・ロジック上の判定問題として,ここでは,論理式の恒真性判定について考察する.テンポラル・ロジックの論理式Pがあるモデル(Σ, M)の上で恒真であるとは,Σ上の任意の系列xに対しx(P)=Tまたはx(P)=⊥となることである. x(P)=⊥は,系列xの長さが論理式Pの真偽値を決定するのに十分長くないことを表わす.よって,論理式が恒真であるかどうかの判定は,$S(P) = (\Sigma^* - \{x \in \Sigma^* \mid x(P) = \bot\})$の判定である.補題1より,x(P)=⊥となる系列xの個数はたかだか有限個であるので,この等式の右辺は正則集合である.正則集合間の包含判定は決定可能であるので,上記の判定は,S(P)が正則集合であれば決定可能であること,およびS(P)がcflのクラスに属すならば決定不能であることがいえる(14).よって,テンポラル・ロジックにおける論理式Pに対する系列集合が正則集合であれば,テンポラル・ロジックにおける真偽判定は決定可能である.

テンポラル・ロジックにおける論理式Pに対する系列集合が正則集合であることを決定する要因としては二つある.一つは原子命題に対する系列集合の対応である.他方は,論理演算に対応する系列集合上の演算が正則集合のクラスで閉じるかどうかである.前者については,4.で述べたように,PTL, ETL, Local ITLにおいては,原子命題に正則集合を割り当てていることがわかっている.また,後者については,テンポラル・ロジックにおける任意の論理演算に対応する系列集合上の演算は正則集合のクラスで閉じることが証明できる.

[定理 2]

テンポラル・ロジックにおける演算に対応する系列集合上の演算は,いずれも正則集合のクラスで閉じる. □

[証明]

¬, ∧および○対応する演算が正則集合のクラスで閉じることは明らかである．また，□, G, ; などに対応している一般化連接は，いずれも，$x=x'z$, $y=zy'$を連接して$x'zy'$を構成するものである．このような一般化連接は，zの属す系列集合をZとすると，$(x/Z)\cdot y \cap x\cdot(Z\backslash y)$と表わすことができる．正則集合に関する左および右からの商は正則集合のクラスで閉じることがわかっている[8]．□, G, ; などに対応している一般化連接では，Zの部分は必ず正則集合であるので，これらの一般化連接は正則集合のクラスで閉じる．

一方，正則集合Xを，n記号重ねる一般化連接により閉包をとった系列集合は，Xを受理する有限オートマトンの最終状態qに，初期状態からqへ至る系列xの先頭n記号を除いた系列$x[n+1, |x|]$で初期状態から到達できる状態から出ている枝を付加した有限オートマトンにより受理される．□

定理2より，つぎの定理が導かれる．

[定理3]

(1) PTL, ETL, Local ITLにおける真偽判定は決定可能である．

(2) ITLにおける真偽判定は決定不能である．□

とくに，ITLにおける決定不能性は，原子命題に正則集合以外の系列集合を割り当てることにより生じる．Local ITLでは，ITLに対し，原子命題に長さ1の系列集合を割り当てるという制限を行うことにより決定可能性を保証しているが，原子命題に対し正則集合のみを割り当てる論理体系(Regular ITLと呼ぶ)においても決定可能性は保証される．ただし，Regular ITLでは，$S \leftrightarrow 0S1$ ($P \leftrightarrow Q$は$\neg(\neg P \wedge Q) \wedge \neg(P \wedge \neg Q)$の略記法である)のような論理式を満たす系列集合をSに割り当てることはできず，論理式の充足性について，ITLとは能力が異なる．以上の結果をまとめると，表1に示すようになる．

6. むすび

本報告では，命題テンポラル・ロジック，拡張テンポラル・ロジックおよびインターバル・テンポラル・ロジックにおける論理演算に対応する系列集合上の演算について述べた．さらに，これらの演算がいずれも正則集合のクラスで閉じていることを示し，テンポラル・ロジック上の論理式の真偽判定の決定可能性に関する種々の結果が，系列集合上の包含判定の決定可能性から自然に導かれることを示した．

今後は，表1における各体系の記述能力の差についてさらに詳しく考察するとともに，論理式の充足可能性と系列集合の関係について考察したい．また，一般化連接そのものの性質についても考察したい．

表1 テンポラル·ロジックと系列集合のクラスの対応

テンポラル·ロジックのクラス	系列集合のクラス	決定可能性	対応する一般化連接
PTL	正則の部分	可	1記号ずらして重ねる
Local ITL	正則の部分	可	1記号重ねる
ETL	右線形文法と表現等価	可	n(≧1)記号ずらして重ねる
Regular ITL (原子命題に正則集合を割り当てる)	正則の部分	可	1記号重ねる
平石の体系 (9)	正則集合と表現等価	可	通常の連接
cfl ITL (原子命題にcflを割り当てる)	cfl	不可	1記号重ねる
csl ITL (原子命題にcslを割り当てる)	csl	不可	1記号重ねる
ITL	0型以上	不可	1記号重ねる

謝辞 本稿の作成において御援助賜った神戸大学羽根田博正教授に心から感謝いたします.また,種々御討論頂いた京都大学平石裕実博士に感謝いたします.

文献

(1) A. Pnueli : "The Temporal Logic of Programs", Proc. of the 18th Annual Symposium on Foundations of Computer Science, pp. 46-57 (Oct. 1977).

(2) S. Owicki and L. Lamport : "Proving Liveness Properties of Concurrent Programs", ACM Transactions on Programming Language and Systems, vol. 4, No. 3, pp. 455-495 (July 1982).

(3) Z. Manna and A. Pnueli : "Verification of Concurrent Programs : The Temporal Framework", The Correctness Problem in Computer Science (R. S. Boyer and J. S. Moore, eds.), International Lecture Series in Computer Science, Academic Press (1981).

(4) P. Wolper : "Temporal Logic Can Be More Expressive", Proc. of the 22nd Annual Symposium on Foundations of Computer Science, pp. 340-348 (1981).

(5) G. V. Bochmann : "Hardware Specification with Temporal Logic: an Example", IEEE Transactions on Computers, vol. C-31, no. 3, pp. 223-231 (March 1982).

(6) B. C. Moszkowski : "Reasoning about Digital Circuits", Technical Report STAN-CS-83-970, Stanford University (July 1983).

(7) 藤田,田中,元岡:"ハードウェア状態遷移表現のPrologによる検証",情報処理学会論文誌,第25巻第4号, pp. 647-654 (1984-07).

(8) ホップクロフト,ウルマン(野崎,木村 訳):"言語理論とオートマトン",サイエンス社(1971).

(9) 平石,矢島:"λフリー正則集合に表現等価なテンポラルロジック",1986年冬のLAシンポジウム(1986-01).